

## Palvelutarjonta tietoliikenneverkossa

### Keksinnön ala

- 5 Keksintö liittyy yleisesti ottaen palvelutarjontaan tietoliikenneverkossa. Tarkemmin sanottuna, keksintö liittyy menetelmään ja järjestelmään, jolla tarjotaan palveluja, joihin verkonkäyttäjät pääsevät lähettämällä palvelupyyntöjä luonnollisella kielellä.

### Tekniikan tausta

- 10 Internetin käyttäjien määrä ja internetissä palveluja tarjoavien määrä on kasvanut valtavasti. Tämä on ollut merkittävin ilmiö tietoliikenteessä viime vuosina. Toinen huomattava suuntaus on erilaisten liikkuvien päätteiden, kuten kannettavien tietokoneiden, PDA-laitteiden (Personal Digital Assistant) ja älypuhelinien käytön voimakas kasvu.

- 15 Nämä kaksi nopeasti kehittyvää verkkoteknologiaa, langaton tietoliikenne ja internet, ovat vähitellen lähestymässä toisiaan, jolloin internetissä käytetyt pakettivälitteiset tietopalvelut ovat myös matkapuhelinkäyttäjien ulottuvilla. Koska markkinoille ilmestyy yhä monimutkaisempia järjestelmiä ja palveluja, tarve saada entistä käyttäjäystävällisimpiä ihmisen-kone-liityntöjä on tulossa ensiarvoisen tärkeäksi palvelujen hyödyntämiseksi tehokkaasti.

- 20 Tällä hetkellä useimmat ihmisen-kone-liitynnät ovat suhteellisen monimutkaisia ja vaikeita käyttää. Esimerkiksi tyypillisessä matkaviestinpalvelussa käyttäjän on muistettava palvelurajapinnassa tarvittava semantiikka ja syntaksi. Tällainen järjestelmä ei salli virheitä; jopa pienet kirjoitusvirheet hankaloittavat palvelun käyttöä. Tämän vuoksi on toivottavaa, että käyttäjä voisi ilmaista käskynsä tai kyselynsä niin luonnollisella tavalla kuin mahdollista, käyttämällä luonnollista kieltä.

- 25 Tämän ongelman ratkaisemiseksi on markkinoille tuotu luonnollisen kielen rajapintoja, jotka sallivat käyttäjän ilmaista aikomuksensa luonnollisella kielellä. Tunnettu nettisivu, jossa käyttäjä voi etsiä tietoa englanniksi on "Ask Jeeves service" ([www.askjeeves.com](http://www.askjeeves.com)). Tämä palvelu perustuu luonnollisen kielen tulkinnan ja sovelluskohtaisen tietämyskannan yhdistelmään. Kysymyksen tarkka luonne analysoidaan ensin käyttämällä sekä semanttista että syntaktista prosessointia ja vastaukset etsitään tietämyskannasta, joka
- 35 sisältää tietoa yleisimmin esitetyistä kysymyksistä.

Toista tunnettua luonnollisen kielen tulkintaan perustuvaa järjestelmää selitetään PCT-patenttijulkaisussa WO 99/66381, joka esittelee adaptiivista agenttipohjaista ihminen-kone-rajapinnan ohjelmistoarkkitehtuuria. Tässä järjestelmässä luonnollisen kielen viesti syötetään ohjelmistoagenttien verkkoon. Yksi tai useampi agenteista toimii luonnollisen kielen viestin vastaanottavana syöteagenttina. Verkon eri agentit ovat "hyperkytkettyjä" toisiinsa, mikä tarkoittaa sitä, että ne vastaanottavat kyselyitä ja käskyjä ainoastaan ennalta määrätyiltä verkon toisten agenttien osajoukolta, ja välittävät kyselyt ja käskyt ainoastaan verkon agenttien toiseen osajoukkoon.

10 Nykyisten luonnollisen kielen tulkintaan perustuvien järjestelmien huomattavin haittapuoli on se, että käyttäjäsyötteen tulkinta on tiukasti sidoksissa muun järjestelmän toteutukseen. Toisin sanoen, tulkinta on tehty toimivaksi ainoastaan palvelulle, jota muu osa järjestelmästä tarjoaa. Tämän vuoksi palveluntarjoaja, joka tarjoaa laaja-alaisia palveluja yksittäisen palvelukehityksen puitteissa, ei ole voinut hyödyntää luonnollisen kielen tulkintaa tehokkaasti palveluja tarjotessaan.

Keksinnön päämääränä on saada aikaan ratkaisu, jonka avulla voidaan poistaa yllämainitut ongelmat sekä mahdollistaa erilaisten palvelujen helppo käyttö ja uusien palvelujen ja palveluyhdistelmien tehokas tarjonta.

20

### **Keksinnön lyhyt yhteenveto**

Tämä ja muut keksinnön kohteet toteutetaan esillä olevan keksinnön periaatteiden mukaisesti tarjoamalla järjestelmä, jossa käyttäjä pääsee moniin eri palveluihin käyttämällä palvelupyynnöissä luonnollista kieltä. Keksinnön ajatuksena on erottaa luonnollisen kielen tulkinta palvelutoteutuksesta ja -suorituksesta hyödyntämällä välikerrosta, joka generoi ensin käyttäjän syötteestä geneeristä muotoa olevan palvelukuvauksen ja parsii (jäsentää) sitten tämän kuvauksen informaatioksi, joka osoittaa suoritettavat palvelukutsut ja niihin liittyvät parametrit. Tämä informaatio palautetaan palvelulogiikka-

25 komponentille, joka puolestaan toteuttaa palvelun sinänsä tunnetulla tavalla suorittamalla mainitut palvelukutsut. Luonnollisen kielen syöte muunnetaan siis ensin palvelukuvaukseksi, joka kuvaa luonnollisen kielen viestin tarvitsemaa palvelutoiminnallisuutta. Tämä palvelun kuvaus annetaan yleistetyllä merkkauksielellä (generalized markup language). Kuvaus parsitaan tämän

30 jälkeen elementteihin, jotka identifioivat suoritettavat operaatiot (kuten palvelukutsut) ja näihin operaatioihin liittyvät parametrit, jolloin palveluja

lukutsut) ja näihin operaatioihin liittyvät parametrit, jolloin palveluja voidaan tarjota tavanomaisella tavalla.

Tällä tavalla luonnollisen kielen tulkintaa voidaan käyttää palvelujen tarjoamisessa tehokkaalla tavalla niin, että vasteena käyttäjän syötteeseen voidaan tarjota mittava valikoima palveluja ja palvelujen mielivaltaisia yhdistelmiä. Käyttäjä voi siis saada mittavan valikoiman uusia ja jo olemassa olevia palveluja, ilman että hänen täytyisi muistaa jokaisen palvelun rajapinnan tarvitsema semantiikka ja syntaksi. Palveluntarjoaja, joka tarjoaa useita erillisiä palveluja voi tällä tavalla tarjota uusia palveluja yhdistämällä niitä tavalla, joka määräytyy käyttäjän syöteviestin perusteella.

### Kuvioluettelo

Seuraavassa keksintöä ja sen edullisia toteutustapoja selostetaan yksityiskohtaisemmin oheisten kuvioissa 1 – 7 kuvattujen esimerkkien avulla, joissa

- kuvio 1 esittää esillä olevan keksinnön mukaisen järjestelmän yleistä arkkitehtuuria,
- kuvio 2 esittää palvelupyyntöviestiä, jonka käyttäjä syöttää päätteeltään,
- 20 kuvio 3 kuvaa esimerkkiä järjestelmän elementtien välisestä sanomavaihdosta, joka tapahtuu vasteena kuvion 2 palvelupyyntöön,
- kuvio 4 esittää keksinnön mukaisen luonnollisen kielen tulkinnan toiminnallista rakennetta,
- kuvio 5 on esimerkki luonnollisen kielen tulkinnasta tulevasta tulosteesta,
- 25 kuvio 6 on vuokaavio, joka kuvaa XML-parserin toimintaa ja
- kuvio 7 esittää palvelulogiikkakomponentin toimintaa sen tarjotessa palveluja vasteena XML-parserista tulevalle syötteelle.

### Keksinnön yksityiskohtainen selostus

30 Kuvio 1 esittää esillä olevan keksinnön mukaisen järjestelmän arkkitehtuuria pääpiirteittäin. Järjestelmä käsittää useita palvelimia, jotka tarjoavat palveluja järjestelmän käyttäjille. Kaksi näistä palvelimista (S1 ja S2) on esitetty kuviossa. Palvelimet tarjoavat verkon käyttäjille mittavan valikoiman resursseja ja palveluja, ja yksi tai useampi palveluntarjoaja voi omistaa palvelimet. Tässä esimerkissä kaikki palvelimet ovat internetin tai intranetin (tai vastaavan TCP/IP-verkon) solmuja. Tässä asiayhteydessä termillä "palvelin"

tarkoitetaan palvelimessa olevaa palvelusovellusta. Palveluja tarjoavia palvelimia kutsutaan jäljempänä myös palvelukehykseksi.

Käyttäjien päätteistä (UT1 ja UT2) on pääsy palveluja tarjoaviin palveluihin sinänsä tunnetulla tavalla. Käyttäjien päätteet voivat olla kiinteitä, 5 kuten on esitetty päätteen UT1 osalta, tai liikkuvia päätteitä, joista on langaton pääsy järjestelmään tai verkkoon, kuten on esitetty päätteen UT2 osalta. Langaton liittymä voidaan toteuttaa eri liittymäpisteiden (AP1) välityksellä tai vaihtoehtoisesti yhdyskäytävien (GW1) välityksellä, jotka muuntavat päätteen ja yhdyskäytävän välillä käytettävän protokollan, kuten WAP:n, pyynnöt protokollaksi, jota käytetään yhdyskäytävän ja palvelimen välillä, toisin sanoen 10 HTTP:ksi.

Keksinnön kannalta ei käyttäjien päätteiden tyypillä eikä päätteiden ja palveluja tarjoavien palvelimien välisten yhteyksien tyypeillä ole merkitystä. Tässä yhteydessä päätteiden ainoa olennainen piirre on se, että ne on 15 varustettu selaimilla tai muilla tunnetuilla asiakasohjelmilla, joiden avulla ne voivat kommunikoida palveluja tarjoavien verkon palvelimien kanssa.

Järjestelmä käsittää edelleen palvelulogiikkakomponentin SL, luonnollisen kielen tulkin NLI ja parserin PS. Nämä kolme yksikköä muodostavat esillä olevan keksinnön mukaisen järjestelmän ytimen. Tämän ytimen 20 avulla käyttäjät voivat tehokkaasti käyttää palvelukehyksen palveluja syöttämällä luonnollisen kielen palvelupyynnöjä. Palvelulogiikkakomponentti vastaanottaa palvelupyynnöt käyttäjien päätteiltä, toimittaa ne edelleen luonnollisen kielen tulkille, vastaanottaa tulkilta tulokset ja toimittaa ne edelleen parserille prosessoitaviksi. Palvelulogiikkakomponentti vastaanottaa 25 parsinnan tulokset parserilta ja vasteena näihin suorittaa ne oliot, jotka kutsuvat toivottuja palveluja palvelukehyksessä. Palvelulogiikkakomponentti toimii näin ollen kuin napana: se koordinoi ensin palvelupyynnön tuloksin ja parsimisen ja toteuttaa sen jälkeen palvelun parsittujen tulosten perusteella. Luonnollisen kielen tulkitsee käyttäjän luonnollisen kielen syötteeseen ja 30 muuntaa tulkitut tulokset geneeristä muotoa olevaksi palvelukuvaukseksi, joka kuvaa käyttäjän pyynnön täyttämiseen tarvittavaa palvelun toiminnallisuutta.

Kuvio 3 esittää esimerkin sanomavaihdosta järjestelmän elementtien välillä, silloin kun käyttäjä käyttää palvelukehyksessä olevaa palvelua. Tässä yhteydessä käytetään yhteystietoja tarjoavaa palvelua esimerkkinä 35 kuvaamaan esillä olevan keksinnön mukaista palveluntarjontaa.

Ensiksi käyttäjä antaa syötekomennon tai -viestin luonnollisella kielellä näppäilemällä tai syöttämällä halutun käskyn tai viestin muulla tavalla omalta päätteeltään. Kuvio 2 kuvaa esimerkkiä viestistä, jonka käyttäjä voi syöttää päätteeseen löytääkseen tietyn henkilön yhteystiedot (esim. Jaakko Väisänen). Käyttäjä lähettää viestin palvelulogiikalle SL (vaihe 301 kuviossa 3), joka välittää viestin luonnollisen kielen tulkille NLI (vaihe 302). Tulkki analysoi käyttäjän syöttämää luonnollisen kielen viestiä (vaihe 303) ja lähettää palvelukuvauksen yleistetyllä merkkauksielellä, edullisesti XML-muodossa. XML (eXtensible Markup Language) on standardisoitu kieli, joka on luotu helpottamaan rakenteisten dokumenttien ja tietojen kuvausta ja käyttöä verkossa. Itse tulkinta suoritetaan tunnetulla tavalla. Palvelukuvauksen geneerinen muoto käsittää palvelutoiminnallisuuksien nimet, joita tarvitaan täyttämään käyttäjän toiveet, ja parametrit (käyttäjän syöttämät), joita mainitut toiminnallisuudet tarvitsevat. Palvelukuvaus palautetaan sitten palvelulogiikalle vaiheessa 304. Yleisesti ottaen palvelukuvaus määritellään XML-sovelluksena.

Palvelulogiikka lähettää sitten parserille, joka on nyt XML-parseri, pyynnön parsia tulkilta saatu XML-kuvaus (vaihe 305). Pyyntö käsittää XML-muotoa olevan jonon, joka on saatu tulkilta. Parseri analysoi jonon ja instantioi jonosta löytyneiden identiteettien perusteella löydettyjä identiteettejä vastaavat luokat. Luotuja ilmentymiä kutsutaan olioiksi, kuten on yleistä oliopohjaisessa ohjelmoinnissa. Oliot käsittävät ohjelmakoodin, joka vastaa niitä toiminnallisuuksia, jotka on määritelty XML-kuvauksessa (vaihe 306). Tässä esimerkissä parseri löytää XML-kuvauksesta sen toiminnallisuuden nimen, joka kuvaa yhteystietoja tietokannasta hakevan toiminnon (vaihe 306). Vaiheessa 307 parseri palauttaa vastaavat oliot palvelulogiikalle. Palvelulogiikka tallentaa oliot siinä järjestyksessä, joka vastaa mainittujen toiminnallisuuksien suoritusjärjestystä. Tiettyä palvelutoiminnallisuutta tarjoava olio käsittää tyypillisesti palvelupyynnön lähettämisen kyseessä olevaa palvelua tarjoavalle palvelimelle (vaihe 309) ja vasteen generoimisen käyttäjälle hänelle palautettavista palvelutuloksista (vaihe 311). Palvelulogiikka suorittaa yhden olion kerrallaan vaiheessa 308. Kuviossa 3 esitetään ainoastaan yhden olion suorittaminen, toisin sanoen kuvio 3 esittää ainoastaan yhden palvelupyynnön/vasteparin.

Parseri instantioi siis niitä luokkia, jotka suorittavat XML-kuvauksessa osoitettuja toiminnallisuuksia. Nämä ilmentymät (eli oliot) palau-

tetaan palvelulogiikalle SL, joka puolestaan suorittaa toiminnallisuudet kutsumalla objekteja oikeassa järjestyksessä.

Palvelutarjonnan toiminnallisuutta esitellään seuraavassa yksityiskohtaisemmin esittelemällä järjestelmälle ominaisia elementtejä.

- 5 Kuviossa 4 esitetään luonnollisen kielen tulkin funktionaalista rakennetta. Tulkki käsittää agenttien muodostaman verkon tai agenttiyhteisön. Kuviossa jokainen ovaali edustaa yhtä agenttia. Käyttäjän luonnollisen kielen viesti syötetään syöteagentille 41, viesti on tässä esimerkissä samanlainen kuin kuviossa 2. Verkon eri agentit ovat yhteydessä toisiinsa siten, että yksit-
- 10 täinen agentti vastaanottaa kyselyjä tai käskyjä vain ennalta määrätyltä toisten agenttien osajoukolta ja siirtää kyselyjä tai käskyjä vain toiselle agenttien osajoukolle. Jokaisella agentilla on vastuullinen hallinta-alue luonnollisen kielen viestin tulkitsemisessa. Jokaiselle järjestelmän tarjoamalle toiminnallisuudelle (eli palvelulle) on määrätty siitä palvelusta vastuussa oleva agentti, ja
- 15 jokainen agentti on ensisijassa vastuussa vain yhdestä palvelusta. Lisäksi osa agenteista on vastuussa palvelun tarvitsemien parametrien tulkitsemisesta. Yksi agenteista on esimerkiksi yhteystietoagentti (42), joka on vastuussa yhteystietopalveluista. Tämä agentti vastaanottaa kaikki viestit, jotka käsittävät sanoja, jotka viittaavat yhteystietopalveluun, kuten esim. "yhteys-
- 20 tiedot" (contact information), "puhelinnumero" (telephone number) jne. Lisäksi yksi agenteista voi olla kokonimiagentti (43), joka on vastuussa niistä parametreista, joita yhteystietopalvelu tarvitsee ja joka siis vastaanottaa ja prosessoi niiden henkilöiden nimiä, joiden yhteystietoja etsitään.

- Verkkoagentit voidaan toteuttaa samalla tavalla kuin yllämainituksa PCT-hakemuksessa WO 99/66381 on kuvattu, ja siksi siihen ei paneuduta
- 25 tässä sen tarkemmin. Mainitussa WO-julkaisussa kuvatun toteutuksen lisäksi yksittäiset agentit on suunniteltu sellaisiksi, että ne generoivat tulkintatuloksista XML-muotoisen kuvauksen. Yhteystietoagentti generoi XML-osioita, jotka osoittavat kyseessä olevan toiminnon yhteystietopalveluksi ja kokonimi-
- 30 agentti generoi XML-muotoiset parametrit sen henkilön nimen perusteella, jonka yhteystietoja etsitään. Kaikki generoidut XML-kuvaus-elementit siirretään XML-sääntöagentille 44, jolla on asiaan kuuluva XML-sääntömäärittely. Käyttämällä tätä määrittelyä hyväkseen XML-sääntöagentti asettaa generoidut elementit oikeaan järjestykseen niin, että saadaan halutun palvelun XML-
- 35 kuvaus. Tämä kuvaus on siis generoitu käyttäjän syöttämän luonnollisen kielen syötteen perusteella.

Ohjelmistoagenttiverkon sijasta tulkinnassa voidaan käyttää myös muita ratkaisuja. Järjestelmän arkkitehtuuri ei aseta mitään vaatimuksia tulkin rakenteelle, lukuun ottamatta sitä, että tuloste on esitetty yleistetyllä merk-  
kauskielellä, edullisesti XML-kielellä.

5           Kuviossa 5 on esimerkki XML-kuvauksesta, joka on saatu kuvioi-  
den 2 ja 4 mukaisen käyttäjän syötteen tuloksena. Kuviossa 5 ensimmäinen  
rivi käsittää XML-deklaraation ja toinen rivi dokumentin tyyppin deklaraation  
(esittelyn). Kolmas rivi käsittää juurielementin, joka sisältää dokumentin kaik-  
ki muut elementit. Kuten tunnettua, jokainen elementti käsittää alkutagin, lop-  
10 putagin ja tagien välillä olevan sisällön. Neljäs rivi osoittaa kyseessä olevan  
toiminnallisuuden nimen "contact\_search", rivi on kuvion 4 mukaisen yhteys-  
tietoagentin generoima. Viides rivi osoittaa, että toiminnallisuudella on siihen  
kuuluva parametri, jonka nimi on "ContactFullName" ja jonka arvo tässä ta-  
pauksessa on "Jaakko Väisänen". Kuvion 4 kokonimiagentti on generoinut  
15 tämän rivin.

Tässä vaiheessa luonnollisen kielen tulkki on siis tulkinnut alkupe-  
räisen luonnollisen kielen niin, että kyseessä oleva toiminnallisuus on "con-  
tact\_search", joka on järjestelmässä oleva yhteystietopalvelu, ja että etsittävä  
yhteystieto on se, jonka parametrilla "ContactFullName" on arvo "Jaakko Väi-  
20 sänen". Tulkitut tulokset ovat geneerisessä XML-muodossa, kuten kuviossa 5  
on esitetty.

Kuvio 6 on vuokaavio, joka esittää XML-parserin toimintaa. Parseri  
vastaanottaa ensin XML-muotoisen palvelukuvauksen, esim. sellaisen kuin  
on esitetty kuviossa 5, palvelulogiikkakomponentilta (vaihe 610). Parseri ja-  
25 kaa vastaanotetun jonon XML-elementteihin (vaihe 612) jonoissa olevien ta-  
gien mukaisesti, toisin sanoen tagit jakavat jonon elementteihin. Yksi ele-  
mentti käsittää osat, jotka sijaitsevat käskytyyppisen tagin (<command...>) ja  
seuraavan käskyn lopettavan tagin (</command>) välissä. Kuvion 5 mukai-  
sessa esimerkissä parseri löytää siis ainoastaan yhden elementin, joka käsit-  
30 tää rivien 4 – 7 osiot.

Elementit, jotka käsittävät toiminnallisuuksien nimet ja niihin kuu-  
luvien parametrien arvot, parsitaan ja jokaisen elementin tyyppi tutkitaan vai-  
heessa 612 elementti kerrallaan. Kun ensimmäisen elementin tyyppi on mää-  
ritetty, parseri instantioi löydettyä tyyppiä vastaavan luokan (vaihe 613). Täs-  
35 sä vaiheessa parseri siis kuvaa XML-muotoisen kuvauksen luokaksi, joka  
edustaa palvelulogiikkakomponentilla suoritettavissa olevaa palvelutoteutus-



ta. Tätä tarkoitusta varten parseri voi käyttää mitä tahansa kehitysympäristön tarjoamaa kuvausmenetelmää (esim. refleksiivisyyttä (reflection) Javassa). Jokainen kuvattu XML-elementin tyyppiattribuutti on luokka ja jokainen tyyppiattribuutti vastaa yhtä luokkaa. Yksi luokka vastaa yhtä toiminnallisuutta.

5 Parseri luo yhden ilmentymän jokaisesta luokasta, joka on löydetty tyyppiattribuuttien perusteella. Kuten edellä on mainittu, luokkien ilmentymiä kutsutaan tässä yhteydessä olioiksi.

Tämän jälkeen parseri tarkistaa, jäikö parsittuun XML-jonoon elementtejä (vaihe 614). Jos näin on, järjestelmä palaa vaiheeseen 612 tutkiakseen seuraavan elementin tyyppin. Kun kaikkia löydettyjä tyypppejä vastaavat luokat on instantioitu, parseri lähettää parsinnan tulokset palvelulogiikalle SL (vaihe 615). Parsintatulokset käsittävät ne oliot, jotka vastaavat XML-kuvauksessa kuvattuja toiminnallisuuksia. Nämä oliot käsittävät lisäksi jokaiseen toiminnallisuuteen liittyvien parametrien arvot.

15 Kuviossa 7 kuvataan palvelulogiikan SL toimintaa sen vastaanottaessa parsitun vastauksen XML-parserilta. Kuvion vasen puoli osoittaa palvelulogiikan toiminnan vuokaavion muodossa. Vuokaavion ympyrät edustavat niitä kohtia, joista viestit lähetetään verkkoon. Kun palvelulogiikka SL on vastaanottanut parserilta tulevat oliot (vaihe 710), se tallettaa ne suoritusjärjestyksessä (vaihe 711). Tämän jälkeen palvelulogiikka SL kutsuu ensimmäisen olion suoritusmetodia vaiheessa 712. Jokaisella luokalla on oma suoritusmetodinsa vastaavien toiminnallisuuksien toteuttamiseksi. Kun tätä metodia kutsutaan, suoritetaan kyseessä olevan luokan mukainen toiminnallisuus. Toisin sanoen, palvelukäsky lähetetään kyseistä palvelua tarjoavalle palvelimelle (vaihe 309 kuviossa 2). Kun vastaus (esim. tässä tapauksessa Jaakko Väisäsen yhteystiedot) vastaanotetaan yhteystietopalveluja tarjoavalta palvelimelta Sn (vaihe 310 kuviossa 3), palvelulogiikka SL tarkistaa, onko olioita vielä jäljellä (vaihe 714). Jos näin on, palvelulogiikka prosessoi jäljellä olevat oliot samalla tavalla, toisin sanoen prosessi palaa vaiheeseen 712 kutsuakseen seuraavan olion suoritusmetodia. Yleisesti ottaen, kun kaikki metodit on suoritettu, palvelulogiikka SL muuntaa jokaisesta suoritetusta oliosta generoidut vastaukset käyttäjän päätteen tyyppille soveltuvaan muotoon. Tyypillisimmillään palvelulogiikka SL generoi HTML-sivun vastaanotettujen tulosten perusteella ja lähettää sen käyttäjälle (vaihe 715 kuviossa 7 ja vaihe 311 kuviossa 3).

20

25

30

35

Edellä kuvatun suoraviivaisen suorituksen lisäksi toiminnallisuus voi tarvita käyttäjän vuorovaikutusta yllä kuvatun prosessin keskellä. Tämä voi tulla kyseeseen siinä tapauksessa, että sillä hetkellä suorituksessa oleva olio huomaa, että sillä ei ole kaikkea tarvittavaa tietoa toiminnallisuutensa suorittamiseksi, toisin sanoen edelliset oliot eivät ole tuottaneet tarpeeksi tietoa, jotta olio voitaisiin suorittaa täydellisesti loppuun. 5 10

Palvelulogiikka SL vastaanottaa siis vähintään yhden olion parse-  
rilta, tämä olio käsittää XML-palvelukuvauksessa esitetyn toiminnallisuuden  
15 koodin. Vastaanotettuaan oliot ja niihin sidoksissa olevat parametrit palvelu-  
logiikalla SL on täydellinen kuvaus toteutettavasta palvelutoiminnallisuudes-  
ta. Toisin sanoen, palvelulogiikalla on nyt oliojoukko, joka käsittää suoritetta-  
van koodin halutun toiminnallisuuden, kuten henkilön yhteystietojen haun,  
saavuttamiseksi, sekä käytettävät parametriarvot. Sen sijaan, että suoritus-  
20 kelpoinen binääridata palautettaisiin palvelulogiikalle on mahdollista optimoi-  
da järjestelmän toiminta lähettämällä palvelulogiikalle vain viittaukset (toisin  
sanoen osoittimet) olioihin. Itse asiassa kaupalliset välineet, joita voidaan  
käyttää järjestelmän toteuttamisessa, toimivat tyypillisesti tällä tavoin (toisin  
sanoen kaikki muuttujat ja oliot ovat viittauksia).

25 Esimerkkinä ylläkuvatususta toiminnasta voidaan todeta, että yhteys-  
tietopalvelusta vastaavan luokan nimi voisi olla "ContactSearch" ja sen suori-  
tusmetodi voisi olla "ContactSearch.execute()". Tämä suoritustapa voisi  
antaa palvelukäskyn

```

30     serviceResult=ContactBook.searchContact("Jaakko Väisänen"),
    toisin sanoen se voisi kutsua yhteystietopalveluja tarjoavassa palvelimessa
    olevan ContactBook –nimisen palvelun yhteydenhakumetodia syöttäen kut-
    suun XML-parserin parsimia parametreja (t.s. FullName="Jaakko Väisänen").
    Vasteena "ContactSearch" –luokka vastaanottaa ContactBook –palvelun me-
    todilta palvelutuloksia, joista luokka generoi vastauksen käyttäjälle käyttämäl-
35   lä sen omaa metodia "generateUserResponse":

```

```
userResponse=generateUserResponse(serviceResult),
```

jossa serviceResult muodostuu Jaakko Väisäsen yhteystiedoista.

Tämä vastaus renderöidään (tehdään yhteensopivaksi) erityyppisille käyttäjäpääteille palvelulogiikkamenetelmän "SCL.result2Terminal" välityksellä:

5           renderedUserResponse=SCL.result2Terminal(userResponse).

Tämän jälkeen renderöity tulos, joka on tässä tapauksessa Jaakko Väisäsen yhteystiedot sisältävä HTML-sivu, lähetetään käyttäjälle.

Yllä kuvatuissa esimerkeissä oletettiin, että luonnollisen kielen tulkki kykeni tulkitsemaan käyttäjäsyötettä, mikä tarkoittaa sitä, että käyttäjä-  
10 syöte sisälsi tarpeeksi tietoa tulkinnan onnistumiseksi. Jos syötteessä ei ole tarpeeksi tietoa tai jos sitä ei voida tulkita yksiselitteisesti, luonnollisen kielen tulkki käynnistää vuorovaikutuksen, jossa tilanteesta informoiva viesti lähetetään ensin palvelulogiikalle. Palvelulogiikka puolestaan lähettää vuorovaikutuspyynnön käyttäjälle, toisin sanoen prosessi hyppää vaiheesta 303 vaiheeseen 311.  
15

Kuten yllä on kuvattu, keksinnön mukainen järjestelmä käsittää käyttäjän ja palvelujen välisen kerroksen. Tämä kerros generoi ensin käyttäjäsyötteestä geneeristä muotoa olevan palvelukuvauksen ja parsii tämän kuvauksen kokonaisuuksiin, jotka ilmoittavat suoritettavat palvelukutsut ja jokaiseen palvelukutsuun liittyvät parametrit. Keksinnön mukaisesti yhden käyttäjäsyötteen perusteella voidaan suorittaa useita palveluja, koska palvelulogiikka voi kutsua useita palveluja vasteena yhdelle käyttäjäsyötteelle. Toisin sanoen, yksi käyttäjäsyöte ei ole enää sidoksissa täsmälleen yhteen palveluun, vaan vasteena yhdelle käyttäjän luonnollisen kielen syötteelle voidaan tarjota palvelujen yhdistelmiä. Tarjottavien palvelujen yhdistelmä voi olla mielivaltaisen riippuen luonnollisen kielen syötteen sisällöstä. Lisäksi palvelulogiikka voi prosessoida eri palveluista saadut tulokset monella eri tavalla. Tämän vuoksi on mahdollista luoda tehokkaasti uusia räätälöityjä palveluja jo olemassa olevista palveluista.  
20  
25

30           XML-elementtien tyyppiattribuutit voidaan kuvata suoraan palvelukutsuiksi hyödyntämättä edellä kuvattuja luokkia. Tämänkaltaisen toteutus, jossa parseri antaa palvelukutsuja, jotka vastaavat parserin löytämien attribuuttien tyyppejä, on käytännössä toteuttamiskelpoinen ainoastaan yksinkertaisissa toiminnallisuuksissa. Jos toiminnallisuudet ovat monimutkaisempia, sellaisia joissa tarvitaan esim. käyttäjän vuorovaikutusta, on edellä kuvattuja  
35 luokkia hyödyntävä toteutustapa edullisempi. Tämä johtuu siitä tosiasiasta,

- että edellä mainitun käyttäjän vuorovaikutuksen toteuttaminen edellyttää käyttäjän tilan käsittelyä, jota ei ole järkevää toteuttaa ilman yllä kuvattuja luokkia. Tämän lisäksi palveluyhdistelmien tarjoaminen käyttämällä palveluvasteita syöteinformaationa toisille palveluille ja käyttäjäistunnon tilan tarkkaileminen voidaan toteuttaa yleisemmällä ja valvotummalla tavalla, jos luokkia hyödynnetään yllä kuvatulla tavalla.

- Palvelut voidaan integroida järjestelmään monella eri tavalla. Yleisesti ottaen palvelukehykseen kuuluva palvelu voi olla mikä tahansa sisältöä tarjoava moduuli, kuten palvelusovellus tai tietokanta. Palvelusovelluksia voidaan hyödyntää joko käyttämällä palveluohjelmointirajapintaa millä tahansa yhteensopivalla ohjelmointikielellä tai käyttämällä mitä tahansa yleisellä kuvauskielellä, kuten XML:stä sovelletulla WIDL -kielellä (Web Interface Description Language) kuvattua palvelun käyttäjärajapintaa. Palvelusovelluksia voidaan siis hyödyntää myös käyttämällä käyttäjärajapinnan XML-kuvausta.
- 15 Erilaisia adapttereita voidaan integroida palveluintegrointia varten.

Kuten on ilmeistä, eivät keksinnön suoritusmuodot ole rajoittuneet edellä esitettyihin esimerkkeihin ja liitteenä oleviin kuviin, vaan alan ammattilainen voi muokata niitä poikkeamatta keksinnön piiristä ja hengestä.



### Patenttivaatimukset

1. Menetelmä palvelujen tarjoamiseksi tietoliikenneverkossa, jossa menetelmässä

- 5                   - annetaan palvelupyyntö luonnollisella kielellä,  
                  t u n n e t t u siitä, että lisäksi menetelmässä
- tulkitaan palvelupyyntö ja generoidaan palvelukuvaus mainitun tulkinnan perusteella, joka palvelukuvaus on esitetty yleistetyllä merkkauksielellä,
- 10                  - parsitaan palvelukuvaus elementteihin, jotka osoittavat palvelun tarjoamiseksi suoritettavat toiminnallisuudet ja mainittujen toiminnallisuuksien parametriarvot ja
- suoritetaan mainitut toiminnallisuudet käyttämällä mainittuja parametreja.

15                  2. Patenttivaatimuksen 1 mukainen menetelmä, t u n n e t t u siitä, että parsimisvaiheessa:

- kuvataan mainitut elementit luokkiin, jotka toteuttavat mainitut toiminnallisuudet ja
- luodaan olioita mainituista luokista.

20                  3. Patenttivaatimuksen 2 mukainen menetelmä, t u n n e t t u siitä, että suoritusvaihe käsittää mainittujen olioiden suorituksen mainituilla parametriarvoilla.

                  4. Patenttivaatimuksen 1 mukainen menetelmä, t u n n e t t u siitä, että generointivaihe käsittää XML-muotoisen palvelukuvauksen generoinnin.

25                  5. Patenttivaatimuksen 1 mukainen menetelmä, t u n n e t t u siitä, että tulkintavaihe käsittää ohjelmistoagenttiverkon käytön.

                  6. Patenttivaatimuksen 1 mukainen menetelmä, t u n n e t t u siitä, että parsimisvaihe käsittää palvelukuvauksen parsimisen XML-elementteihin.

30                  7. Patenttivaatimuksen 2 mukainen menetelmä, t u n n e t t u siitä, että luontivaihe käsittää niiden luokkien instantioinnin, jotka toteuttavat toiminnallisuuksia.

                  8. Patenttivaatimuksen 3 mukainen menetelmä, t u n n e t t u siitä, että mainittu suoritus käsittää jokaisen parsimisvaiheessa tunnistetun olion suoritusmetodin kutsumisen.

35                  9. Patenttivaatimuksen 1 mukainen menetelmä t u n n e t t u siitä, että se käsittää lisäksi vaiheen, jossa aloitetaan palvelupyynnön lähettäjän kanssa käytävä vuorovaikutus.

10. Palvelutarjontajärjestelmä palvelujen tarjoamiseksi tietoliikenneverkon käyttäjille, joka järjestelmä käsittää

- ainakin yhden sovelluspalvelimen palvelusovellusten tallentamiseksi, jolloin jokaiselle palvelusovellukselle on määrätty operaatio, joka aloittaa palvelusovelluksen, ja

- käyttäjäpäätteitä palvelupyyntöjen syöttämiseksi luonnollisella kielellä,

tunnettu siitä, että järjestelmä käsittää lisäksi

- tulkintavälineet palvelupyynnön tulkitsemiseksi ja yleistetyllä merkkauksielellä olevan palvelukuvauksen generoimiseksi tulkinnan perusteella,

- parsintavälineet, jotka ovat vasteellisia tulkintavälineille, niiden operaatioiden etsimiseksi, jotka palvelukuvauksessa kuvatuille palveluille on määrätty, ja

- suoritusrälineet, jotka ovat vasteellisia parsintavälineille, palvelujen tarjoamiseksi mainittuja operaatioita suorittamalla.

11. Patenttivaatimuksen 10 mukainen järjestelmä, tunnettu siitä, että tulkintavälineet käsittävät ohjelmistoagenttien verkon.

12. Patenttivaatimuksen 11 mukainen järjestelmä, tunnettu siitä, että verkko on sovitettu generoimaan XML-muotoa oleva palvelukuvaus vastena palvelupyyntöön.

13. Patenttivaatimuksen 10 mukainen järjestelmä, tunnettu siitä, että parsintavälineet käsittävät parserin, joka on sovitettu

parsimaan palvelukuvauksen elementteihin, jotka määrittävät ne toiminnallisuudet, jotka on suoritettava palvelun tarjoamiseksi, ja mainittuihin toiminnallisuuksiin liittyvät parametriarvot,

kuvaamaan mainitut elementit luokkiin, jotka toteuttavat mainittuja toiminnallisuuksia, ja

luomaan olioita mainituista luokista.

14. Patenttivaatimuksen 13 mukainen järjestelmä, tunnettu siitä, että suoritusvälineet on sovitettu kutsumaan jokaisen parsintavälineiden luoman olioluokan suoritusmetodia.

15. Patenttivaatimuksen 10 mukainen järjestelmä, tunnettu siitä, että suoritusvälineet käsittävät välineet vuorovaikutuksen aloittamiseksi käyttäjäpäätteiden kanssa.

**Patentkrav**

1. Förfarande för produktion av tjänster vid ett telekommunikationsnät, vid vilket förfarande
- 5           - görs en servicebegäran på ett naturligt språk,  
          k ä n n e t e c k n a t   av att vid förfarandet dessutom  
          - tolkas servicebegäran och genereras en tjänstebeskrivning på basis av nämnda tolkning, vilken tjänstebeskrivning ges på ett generaliserat markeringsspråk,
- 10           - parsas tjänstebeskrivningen till element som anger vilka funktionaliteter skall exekveras för produktion av tjänsten och till vilka parametervärden nämnda funktionaliteter hänför sig, och  
          - nämnda funktionaliteter exekveras med användande av nämnda parametrar.
- 15           2. Förfarande enligt patentkrav 1, k ä n n e t e c k n a t   av att i parsningssteget  
          - mappas nämnda element till klasser som implementerar nämnda funktionaliteter och  
          - skapas objekt ur nämnda klasser.
- 20           3. Förfarande enligt patentkrav 2, k ä n n e t e c k n a t   av att utföringssteget omfattar exekvering av nämnda objekt med nämnda parametervärden.
4. Förfarande enligt patentkrav 1, k ä n n e t e c k n a t   av att genereringssteget omfattar generering av en tjänstebeskrivning i XML-format.
- 25           5. Förfarande enligt patentkrav 1, k ä n n e t e c k n a t   av att tolkningssteget omfattar användning av ett programvaruagentnät.
6. Förfarande enligt patentkrav 1, k ä n n e t e c k n a t   av att parsningssteget omfattar parsning av tjänstebeskrivningen till XML-element.
7. Förfarande enligt patentkrav 2, k ä n n e t e c k n a t   av att skapandet omfattar instansiering av de klasser som implementerar funktionaliteterna.
- 30           8. Förfarande enligt patentkrav 3, k ä n n e t e c k n a t   av att nämnda exekvering omfattar anrop av exekveringsmetoden för varje objekt som har identifierats i parsningssteget.

9. Förfarande enligt patentkrav 1, kännetecknat av att det dessutom omfattar ett steg där sätts i gång interaktion med avsändaren av servicebegäran.

5 10. Tjänsteproduktionssystem för produktion av tjänster för användare av ett telekommunikationsnät, vilket system omfattar

- åtminstone en applikationsserver för lagring av tjänstetillämpningar, varvid för varje tjänstetillämpning har bestämts en operation som startar tjänstetillämpningen, och

10 - användarterminaler för matning av servicebegäran på ett naturligt språk,

kännetecknat av att systemet dessutom omfattar

- tolkningsmedel för tolkning av servicebegäran och för generering av en tjänstebeskrivning på ett generaliserat markeringsspråk på basis av tolkningen,

15 - parsningsmedel som fungerar som respons på tolkningsmedlen för att söka de operationer som har bestämts för de tjänster som beskrivs i tjänstebeskrivningen, och

- exekveringsmedel som fungerar som respons på parsningsmedlen för produktion av tjänsterna genom att utföra nämnda operationer.

20 11. System enligt patentkrav 10, kännetecknat av att tolkningsmedlen omfattar ett programvaruagentnät.

12. System enligt patentkrav 11, kännetecknat av att nätet är anordnat att som respons på en servicebegäran generera en tjänstebeskrivning i XML-format.

25 13. System enligt patentkrav 10, kännetecknat av att parsningsmedlen omfattar en parser som är anordnad att

parsa tjänstebeskrivningen till element som bestämmer vilka funktionaliteter skall utföras för produktion av tjänsten och till vilka parametervärden nämnda funktionaliteter hänför sig,

30 mappa nämnda element till klasser som implementerar nämnda funktionaliteter, och att

skapa objekt ur nämnda klasser.

35 14. System enligt patentkrav 13, kännetecknat av att exekveringsmedlen är anordnade att anropa exekveringsmetoden för varje objekt-klass som parsningsmedlen har skapat.





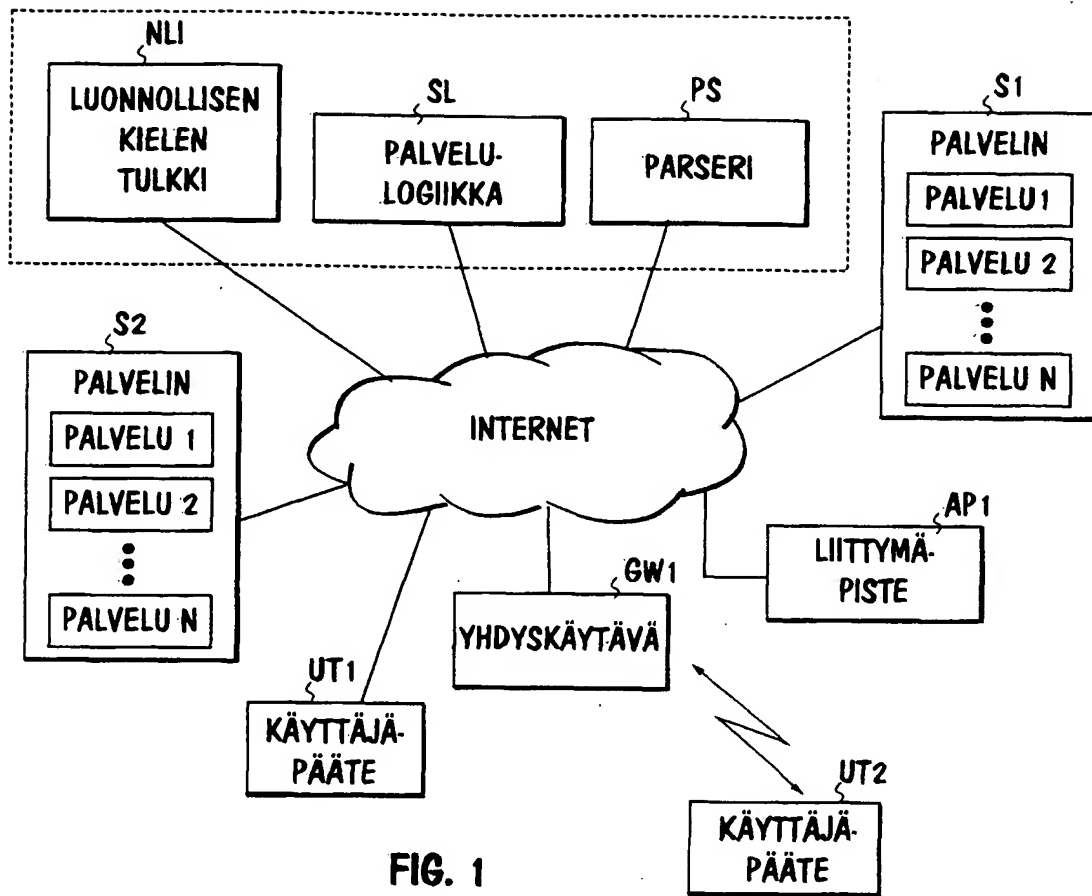


FIG. 1



FIG. 2

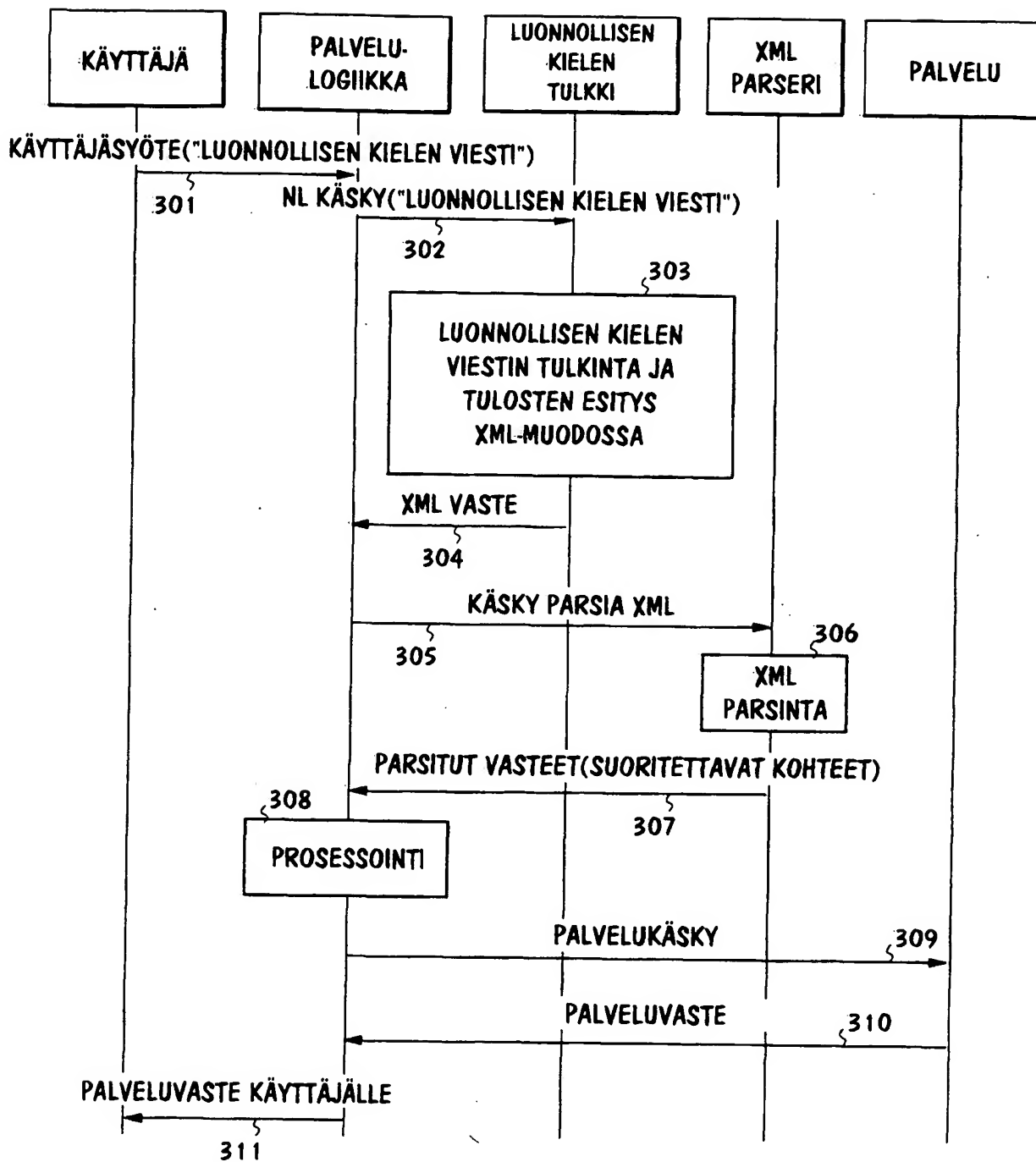


FIG. 3

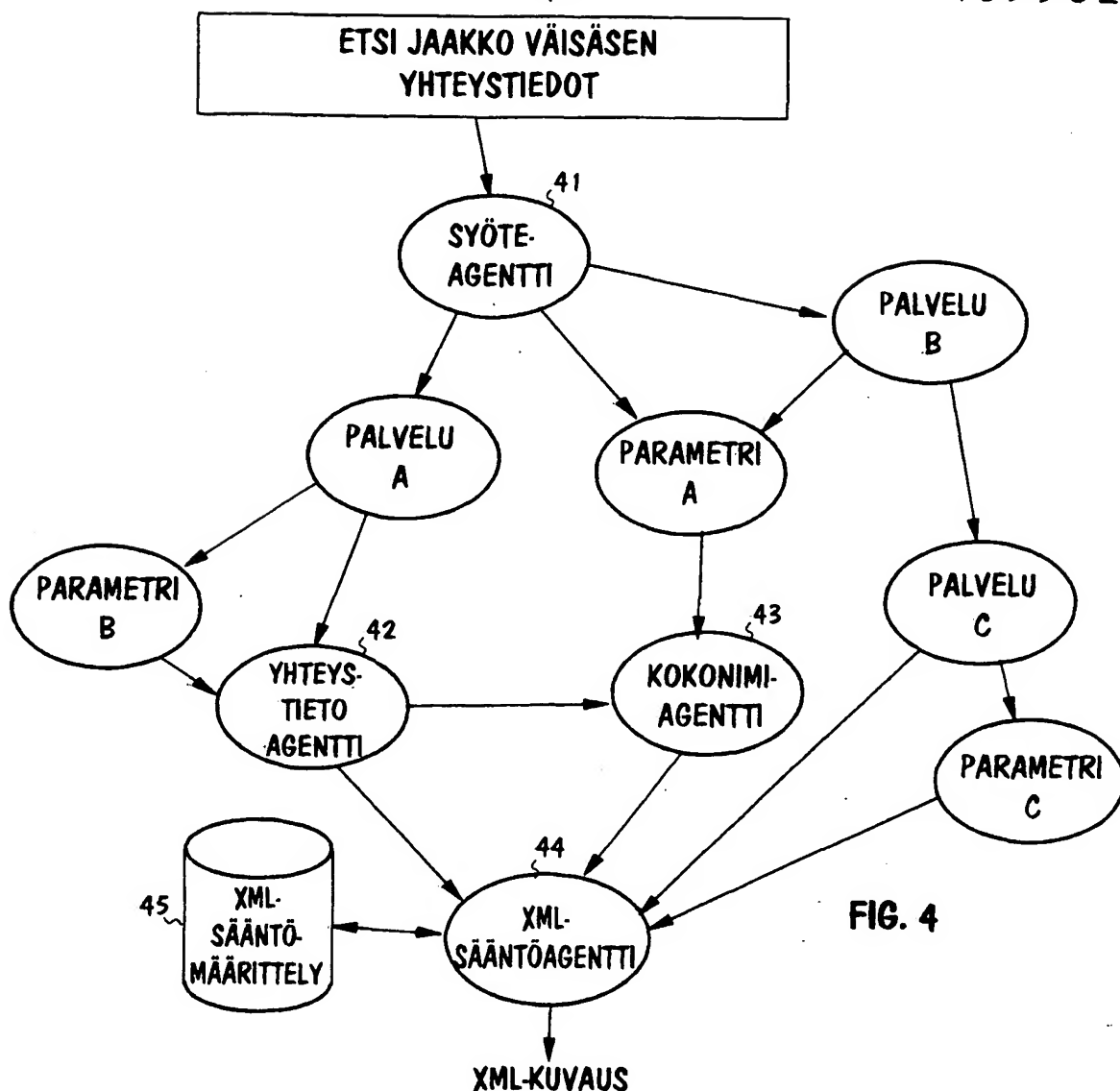


FIG. 4

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE commandresponse SYSTEM "commandresponse.dtd">
<commandresponse>
  <command type="contact_search" index="1">
    <parameter type="ContactFullName">Jaakko Väisänen
  </parameter>
  </command>
</commandresponse>

```

FIG. 5

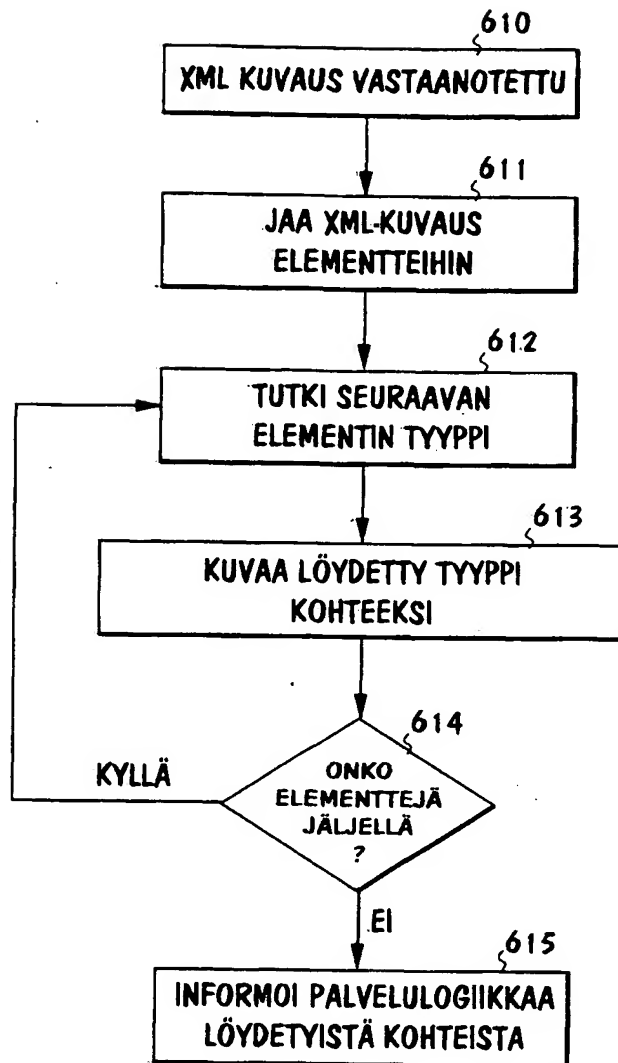


FIG. 6

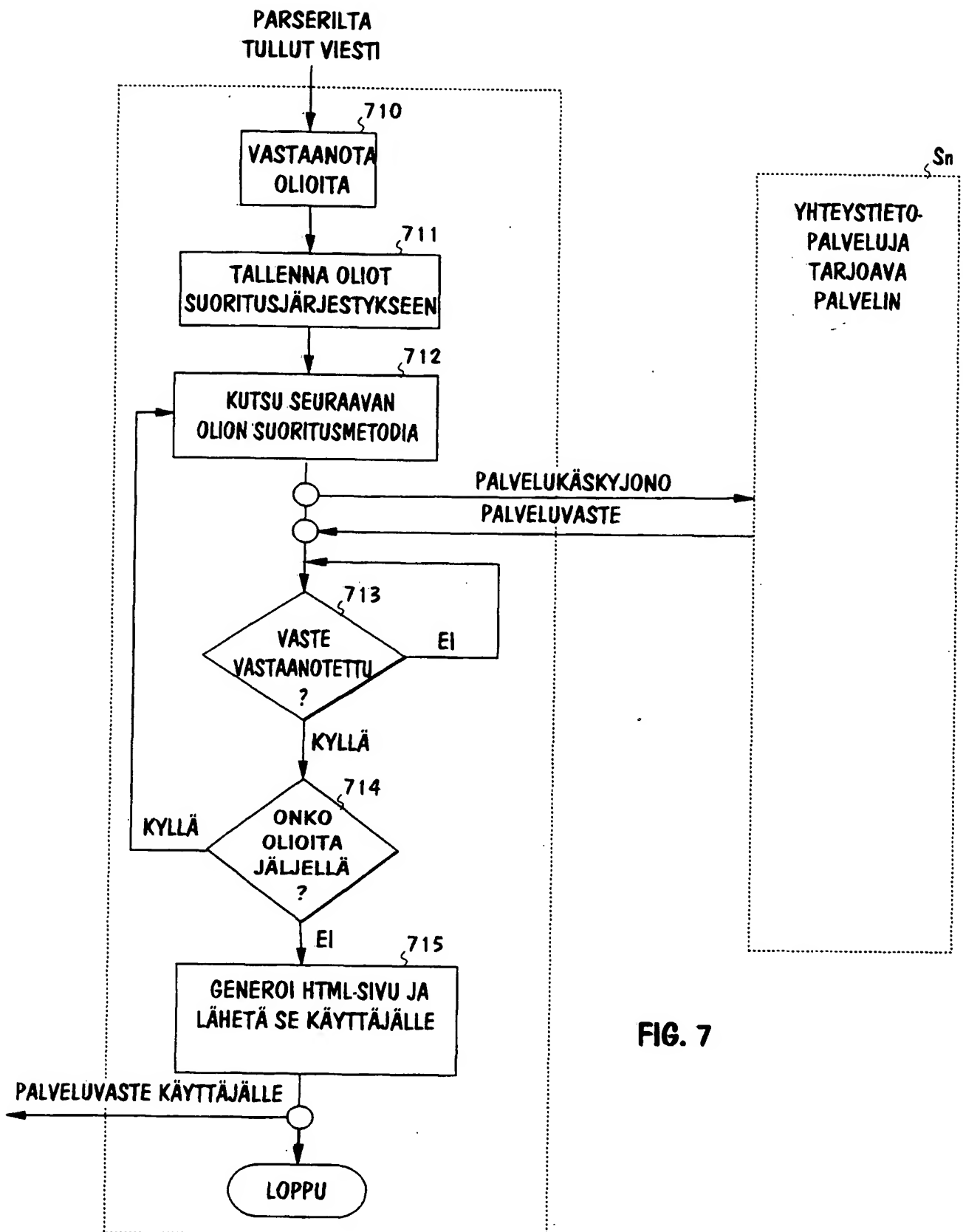


FIG. 7

**PATENTTI- JA REKISTERIHALLITUS**

Patentti- ja innovaatiolinja

PL 1160

00101 Helsinki

**TUTKIMUSRAPORTTI**

|                            |                       |
|----------------------------|-----------------------|
| <b>PATENTTIHAKEMUS NRO</b> | <b>LUOKITUS, IPC7</b> |
| FI 20040296                | H04Q 7/22             |

**TUTKITUT PATENTTILUOKAT (luokitusjärjestelmät ja luokkatiedot)****TUTKIMUKSESSA KÄYTETYT TIETOKANNAT**

EPODOC, WPI, TXTWO1, TXTEP01

**VIITEJULKAISUT**

| Kategoria*) | Julkaisun tunnistotiedot ja tiedot sen olennaisista kohdista | Koskee vaatimusta |
|-------------|--|-------------------|
| A           | FI 109952 B (Sonera Oyj), 7.12.2001                          | 1-18              |
| A           | WO 99/53676 (AT&T Corp.), 21.10.1999                         | 1-18              |
| A           | US 5895466 (AT&T Corp.), 20.4.1999                           | 1-18              |

Jatkuu seuraavalla sivulla

☐

\*) X Julkaisu, jonka perusteella keksintö ei ole uusi tai ei eroa olennaisesti ennestään tunnetusta tekniikasta.  
Y Julkaisu, jonka perusteella keksintö ei eroa olennaisesti ennestään tunnetusta tekniikasta, kun otetaan huomioon tämä ja yksi tai useampi samaan kategoriaan kuuluva julkaisu yhdessä.  
A Yleistä tekniikan tason odustava julkaisu.

O Tulut julkaisoksi esitelmän välityksellä, hyväksikäyttämällä tai muutoin muun kuin kirjoituksen avulla.

P Julkaisu ennen hakemuksen tekemispäivää mutta ei ennen aikaisinta etuoikeuspäivää.

T Julkaisu hakemuksen tekemispäivän tai etuoikeuspäivän jälkeen ja valaisee keksinnön periaatetta tai teoreettista taustaa.

E Aikaisempi suomenlainen tai Suomessa koskeva patentti- tai hyödyllisyysmallihakemus, joka on tullut julkaisoksi hakemuksen tekemispäivänä (etuoikeuspäivänä) tai sen jälkeen.

D Julkaisu, joka on mainittu hakemuksessa.

L Julkaisu, joka kysennäksäänsä etuoikeuden, osoittaa toisen julkaisun julkaisupäivämäärän tai johon viitataan jostakin muusta syystä.

&amp; Samaa patenttiperheeseen kuuluva julkaisu.

Lisätietoja liitteessä

☐

Päiväys Tutkijainvalinnoksi

16.12.2004

Tapio Ikäheimo

## **FI109952B**

Publication Title:

FI109952B

Abstract:

Abstract not available for FI109952B

Abstract of corresponding document: WO0195145

The invention relates to service provision in a communications network. The users of the network can access the services by sending service requests in natural language. The natural language interpretation is separated from the service implementation and execution by utilizing an intermediate layer which first generates a generic format service description out of the user input and then parses this description into information indicating the service calls to be made and the parameters relating to the said calls. This information is returned to a service logic which then implements the service in a manner known as such by making the said service calls.

Data supplied from the esp@cenet database - Worldwide

-----  
Courtesy of <http://v3.espacenet.com>